

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-067905

(43)Date of publication of application : 11.03.1994

(51)Int.Cl.

G06F 9/46

(21)Application number : 04-217611

(71)Applicant : MATSUSHITA ELECTRIC IND CO LTD

(22)Date of filing : 17.08.1992

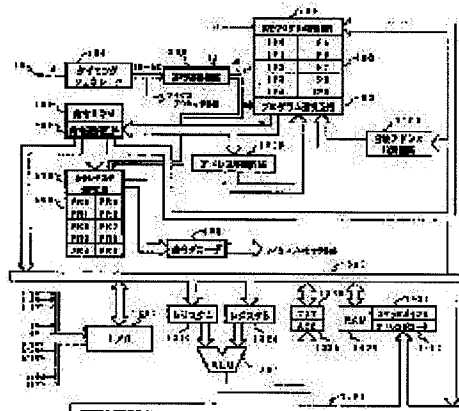
(72)Inventor : SUEHIRO KENICHI  
MIZUGUCHI HIROSHI

## (54) MICROPROCESSOR

## (57)Abstract:

PURPOSE: To generate an execution switching signal corresponding to the number of programs in terms of hardware add to time-divisionally execute the plural programs.

CONSTITUTION: When a processing start instruction is executed, an execution program management circuit 700 sets head address data on the program which is to be newly started in an unused area in instruction pointers IP0-IP9 being the storage areas of an execution address storage circuit 800, and supplies the addition signal of the execution program to an execution switching circuit 200. Thus, the number of the programs to be executed in parallel is increased. When a processing termination instruction is executed, the execution program management circuit 700 sets the instruction pointer where address data of the program terminating execution is stored to be an unused state, and supplies the elimination signal of the execution program to the execution switching circuit 200. Thus, the number of the programs to be executed in parallel is reduced. Thus, the number of the execution programs can easily be changed by software.



(19)日本国特許庁(JP)

(12)公開特許公報(A)

(11)特許出願公開番号

特開平6-67905

(43)公開日 平成6年(1994)3月11日

(51)Int.Cl.<sup>5</sup>

G06F 9/46

識別記号

340 E 8120-5B

庁内整理番号

FI

技術表示箇所

審査請求 未請求 請求項の数6(全21頁)

(21)出願番号 特願平4-217611

(22)出願日 平成4年(1992)8月17日

(71)出願人 000005821

松下電器産業株式会社

大阪府門真市大字門真1006番地

(72)発明者 末廣 憲一

大阪府門真市大字門真1006番地 松下電器  
産業株式会社内

(72)発明者 水口 博

大阪府門真市大字門真1006番地 松下電器  
産業株式会社内

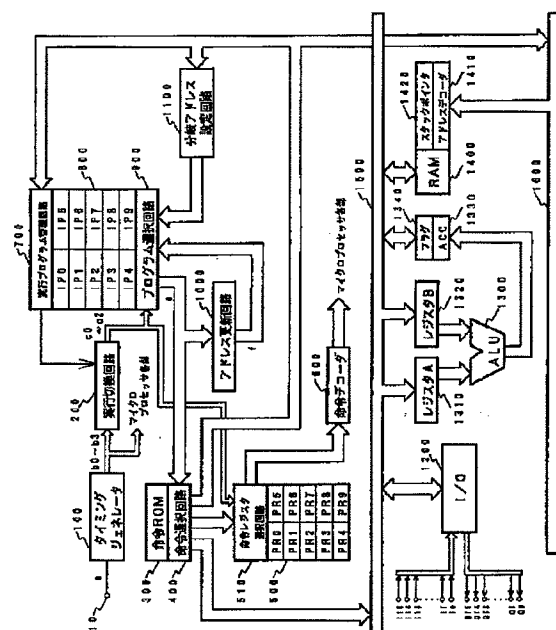
(74)代理人 弁理士 小銀治 明 (外2名)

(54)【発明の名称】 マイクロプロセッサ

(57)【要約】

【目的】 プログラム数に応じた実行切換信号をハードウェア的に発生し、複数のプログラムを時分割で実行できるようにする。

【構成】 処理起動命令が実行されると、実行プログラム管理回路700は、実行アドレス格納回路800の格納領域であるインストラクションポインタIP0～IP9の中の未使用領域に、新たに起動するプログラムの先頭アドレスデータを設定し、実行切換回路200に実行プログラムの追加信号を供給するため、並列に実行するプログラム数が増加する。また、処理終了命令が実行されると、実行プログラム管理回路700は、実行を終了させるプログラムのアドレスデータが格納されたインストラクションポインタを未使用状態とし、実行切換回路200に実行プログラムの削除信号を供給するため、並列に実行するプログラム数が減少する。従って、ソフトウェアにより実行プログラム数を容易に変更できる。



## 【特許請求の範囲】

【請求項1】 命令群で構成された少なくとも1種類以上のプログラムを格納するプログラム格納手段と、供給されるプログラムアドレスデータに応じて、前記プログラム格納手段に格納された特定の命令を選択する命令選択手段と、

実行している各プログラムの実行位置を示すプログラムアドレスデータ群を格納する実行アドレス格納手段と、並列に実行しているプログラムの個数に応じて、プログラムの実行切替信号を出力する実行切替手段と、命令の実行サイクルを発生するタイミングジェネレータと、

前記実行切替信号と前記命令の実行サイクルに基づいて、前記実行アドレス格納手段に格納されているプログラムアドレスデータの何れか1つを選択し、前記命令選択手段に出力する実行プログラム選択手段と、各プログラムの実行に従って、前記実行アドレス格納手段に格納された当該アドレスデータを更新するアドレス更新手段と、

特定のプログラムの実行命令が分岐命令の場合には、前記実行アドレス格納手段の当該実行アドレス格納データを分岐命令が示すアドレスデータに変更する分岐アドレス設定手段と、

特定のプログラムの実行命令が処理起動命令の場合には、新たなプログラムアドレスデータを前記実行アドレス格納手段に追加し、処理終了命令の場合には、実行が終了したプログラムのアドレスデータを前記実行アドレス格納手段から削除する実行プログラム管理手段と、前記プログラム格納手段から出力される命令に基づいてデータの演算を行う演算手段とを備えたマイクロプロセッサ。

【請求項2】 実行アドレス格納手段には最初に起動するプログラムの先頭アドレスがあらかじめ格納されており、初期状態では、このプログラムの実行が自動的に行われることを特徴とする請求項1記載のマイクロプロセッサ。

【請求項3】 実行アドレス格納手段は通常のデジタルデータの格納を行うRAMと共通のデータ格納手段であることを特徴とする請求項1記載のマイクロプロセッサ。

【請求項4】 演算手段は実行するプログラムごとに独立したデータ格納領域を備えていることを特徴とする請求項1記載のマイクロプロセッサ。

【請求項5】 命令群で構成された少なくとも1種類以上のプログラムを格納するプログラム格納手段と、供給されるプログラムアドレスデータに応じて、前記プログラム格納手段に格納された特定の命令を選択する命令選択手段と、

実行している各プログラムの実行位置を示すプログラムアドレスデータ群を格納する実行アドレス格納手段と、

並列に実行しているプログラムの個数に応じて、プログラムの実行切替信号を出力する実行切替手段と、命令の実行サイクルを発生するタイミングジェネレータと、

前記実行切替信号と前記命令の実行サイクルに基づいて、前記実行アドレス格納手段に格納されているプログラムアドレスデータの何れか1つを選択し、前記命令選択手段に出力する実行プログラム選択手段と、

各プログラムの実行に従って、前記実行アドレス格納手段に格納された当該アドレスデータを更新するアドレス更新手段と、

特定のプログラムの実行命令が分岐命令の場合には、前記実行アドレス格納手段の当該実行アドレス格納データを分岐命令が示すアドレスデータに変更する分岐アドレス設定手段と、

特定のプログラムの実行命令が処理起動命令の場合には、新たなプログラムアドレスデータを前記実行アドレス格納手段に追加し、処理終了命令の場合には、実行が終了したプログラムのアドレスデータを前記実行アドレス格納手段から削除する実行プログラム管理手段と、

特定のプログラムの実行命令が連続処理開始命令の場合には、実行切替禁止信号を出力し、連続処理終了命令が実行されるまでプログラムの実行切替を中断させる連続処理制御手段と、

前記プログラム格納手段から出力される命令に基づいてデータの演算を行う演算手段とを備えたマイクロプロセッサ。

【請求項6】 命令群で構成された少なくとも1種類以上のプログラムを格納するプログラム格納手段と、供給されるプログラムアドレスデータに応じて、前記プログラム格納手段に格納された特定の命令を選択する命令選択手段と、

実行している各プログラムの実行位置を示すプログラムアドレスデータ群を格納する実行アドレス格納手段と、前記実行アドレス格納手段のデータ格納状態を示す実行状態フラグと、

並列に実行しているプログラムの個数に応じて、プログラムの実行切替信号を出力する実行切替手段と、命令の実行サイクルを発生するタイミングジェネレータと、

前記実行切替信号と前記命令の実行サイクルに基づいて、前記実行アドレス格納手段に格納されているプログラムアドレスデータの何れか1つを選択し、前記命令選択手段に出力する実行プログラム選択手段と、

各プログラムの実行に従って、前記実行アドレス格納手段に格納された当該アドレスデータを更新するアドレス更新手段と、

特定のプログラムの実行命令が分岐命令の場合には、前記実行アドレス格納手段の当該実行アドレス格納データを分岐命令が示すアドレスデータに変更する分岐アドレ

ス設定手段と、  
特定のプログラムの実行命令が処理起動命令の場合には、新たなプログラムアドレスデータを前記実行アドレス格納手段に追加し、処理終了命令の場合には、実行が終了したプログラムのアドレスデータを前記実行アドレス格納手段から削除する実行プログラム管理手段と、前記プログラム格納手段から出力される命令に基づいてデータの演算を行う演算手段とを備えたマイクロプロセッサ。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は複数のプログラムを時分割で実行することができるマイクロプロセッサに関するものである。

【0002】

【従来の技術】近年マイクロプロセッサはあらゆる方面で多用されており、その構成としては、順次実行される命令群からなるプログラムを格納するプログラム格納手段と、デジタルデータの読み書きが可能なデータ格納手段と、デジタルデータの演算を実行する演算手段と、前記データ格納手段の入出力端子と前記演算手段の入出力端子を接続するデータバスと、前記プログラム格納手段から送出される命令に基づいて前記データ格納手段と前記演算手段の動作をコントロールするコントロール手段と、命令の実行タイミング信号を発生するタイミングジェネレータと、前記タイミングジェネレータの出力に基づいて前記プログラム格納手段に格納された特定の命令を選択する命令選択手段とを備えていることに特徴づけられる。

【0003】

【発明が解決しようとする課題】しかしながら従来の構成では、1つのマイクロプロセッサでは1つのプログラムのみが実行され、さらに、あらかじめ定められた順序に従ってデータの処理を実行していくために、プログラムが膨大になるにつれて、マイクロプロセッサの外部からの要求であり、独立して非同期に発生する外部データの取り込みやそれに基づくデータ処理の実行サイクルが長くなるという問題点を有していた。

【0004】また、データ処理効率を高めるために、メモリ領域や入出力ポートを共有した状態で、複数のマイクロプロセッサを使って、複数のプログラムを並列に実行した場合には、共有したメモリ領域や入出力ポートを全く同時に複数のマイクロプロセッサが使用することによって誤動作が生じるという問題点を有していた。

【0005】さらに、1つのマイクロプロセッサ上で、複数のプログラムの実行の切換を可能にするソフトウェアであるタスクモニタ等を実行し、ソフトウェア的に複数のプログラムを実行した場合には、このタスクモニタ等が、切換を行っている複数のプログラムの管理を行うための複雑なプログラム処理に時間を要するという問題

点や、また、プログラムの実行切換のたびに、演算結果や各種フラグ及び実行を切換える直前のプログラムアドレスデータ等をデータ格納領域に退避させる必要があるため、プログラムの実行切換に時間を要し、切換を頻繁に行うと実際のプログラム処理時間よりもプログラムの実行切換時間の方がマイクロプロセッサの実行時間の大半を占めてしまうという問題点を有していた。

【0006】本発明は上記従来の問題点を解決するもので、ハードウェア的に複数の処理を並列して時分割で実行する、データ処理効率の高いマイクロプロセッサを提供することを目的とする。

【0007】

【課題を解決するための手段】本発明のマイクロプロセッサは、命令群で構成された少なくとも1種類以上のプログラムを格納するプログラム格納手段と、供給されるプログラムアドレスデータに応じて、プログラム格納手段に格納された特定の命令を選択する命令選択手段と、実行している各プログラムの実行位置を示すプログラムアドレスデータ群を格納する実行アドレス格納手段と、並列に実行しているプログラムの個数に応じて、プログラムの実行切換信号を出力する実行切換手段と、命令の実行サイクルを発生するタイミングジェネレータと、実行切換信号と命令の実行サイクルに基づいて、実行アドレス格納手段に格納されているプログラムアドレスデータの何れか1つを選択し、命令選択手段に出力する実行プログラム選択手段と、各プログラムの実行に従って、実行アドレス格納手段に格納された当該アドレスデータを更新するアドレス更新手段と、特定のプログラムの実行命令が分岐命令の場合には、実行アドレス格納手段の当該実行アドレス格納データを分岐命令が示すアドレスデータに変更する分岐アドレス設定手段と、特定のプログラムの実行命令が処理起動命令の場合には、新たなプログラムアドレスデータを実行アドレス格納手段に追加し、処理終了命令の場合には、実行が終了したプログラムのアドレスデータを実行アドレス格納手段から削除する実行プログラム管理手段と、プログラム格納手段から出力される命令に基づいてデータの演算を行う演算手段とを備えたものである。

【0008】また、本発明のマイクロプロセッサは、実行アドレス格納手段には最初に起動するプログラムの先頭アドレスがあらかじめ格納されており、初期状態では、このプログラムの実行が自動的に行われることを特徴とする。

【0009】また、本発明のマイクロプロセッサは、実行アドレス格納手段は通常のデジタルデータの格納を行うRAMと共通のデータ格納手段であることを特徴とする。

【0010】また、本発明のマイクロプロセッサは、演算手段は実行するプログラムごとに独立したデータ格納領域を備えていることを特徴とする。

10

20

30

40

50

【0011】また、本発明のマイクロプロセッサは、命令群で構成された少なくとも1種類以上のプログラムを格納するプログラム格納手段と、供給されるプログラムアドレスデータに応じて、プログラム格納手段に格納された特定の命令を選択する命令選択手段と、実行している各プログラムの実行位置を示すプログラムアドレスデータ群を格納する実行アドレス格納手段と、並列に実行しているプログラムの個数に応じて、プログラムの実行切替信号を出力する実行切替手段と、命令の実行サイクルを発生するタイミングジェネレータと、実行切替信号と命令の実行サイクルに基づいて、実行アドレス格納手段に格納されているプログラムアドレスデータの何れか1つを選択し、命令選択手段に出力する実行プログラム選択手段と、各プログラムの実行に従って、実行アドレス格納手段に格納された当該アドレスデータを更新するアドレス更新手段と、特定のプログラムの実行命令が分岐命令の場合には、実行アドレス格納手段の当該実行アドレス格納データを分岐命令が示すアドレスデータに変更する分岐アドレス設定手段と、特定のプログラムの実行命令が処理起動命令の場合には、新たなプログラムアドレスデータを実行アドレス格納手段に追加し、処理終了命令の場合には、実行が終了したプログラムのアドレスデータを実行アドレス格納手段から削除する実行プログラム管理手段と、特定のプログラムの実行命令が連続処理開始命令の場合には、実行切替禁止信号を出力し、連続処理終了命令が実行されるまでプログラムの実行切替を中断させる連続処理制御手段と、プログラム格納手段から出力される命令に基づいてデータの演算を行う演算手段とを備えたものである。

【0012】また、本発明のマイクロプロセッサは、命令群で構成された少なくとも1種類以上のプログラムを格納するプログラム格納手段と、供給されるプログラムアドレスデータに応じて、プログラム格納手段に格納された特定の命令を選択する命令選択手段と、実行している各プログラムの実行位置を示すプログラムアドレスデータ群を格納する実行アドレス格納手段と、実行アドレス格納手段のデータ格納状態を示す実行状態フラグと、並列に実行しているプログラムの個数に応じて、プログラムの実行切替信号を出力する実行切替手段と、命令の実行サイクルを発生するタイミングジェネレータと、実行切替信号と命令の実行サイクルに基づいて、実行アドレス格納手段に格納されているプログラムアドレスデータの何れか1つを選択し、命令選択手段に出力する実行プログラム選択手段と、各プログラムの実行に従って、実行アドレス格納手段に格納された当該アドレスデータを更新するアドレス更新手段と、特定のプログラムの実行命令が分岐命令の場合には、実行アドレス格納手段の当該実行アドレス格納データを分岐命令が示すアドレスデータに変更する分岐アドレス設定手段と、特定のプログラムの実行命令が処理起動命令の場合には、新たなプ

ログラムアドレスデータを実行アドレス格納手段に追加し、処理終了命令の場合には、実行が終了したプログラムのアドレスデータを実行アドレス格納手段から削除する実行プログラム管理手段と、プログラム格納手段から出力される命令に基づいてデータの演算を行う演算手段とを備えたものである。

#### 【0013】

【作用】上記構成によれば、並列に実行するプログラム数を増加させる場合には、プログラム管理手段が、実行するプログラムのプログラムアドレスデータを実行アドレス格納手段に追加し、実行切替手段が増加した実行プログラム数に応じて、プログラムの実行切替信号を発生する。また、並列に実行するプログラム数を減少させる場合には、プログラム管理手段が、実行を終了させるプログラムのプログラムアドレスデータを実行アドレス格納手段から削除し、実行切替手段が減少した実行プログラム数に応じて、プログラムの実行切替信号を発生する。

【0014】従って、従来のマイクロプロセッサのように独立して非同期に発生する外部データの取り込みやそれに基づくデータ処理の実行サイクルが長くかからず、独立に、しかも非同期に発生する複数の処理に対して即座に対処することができる。

【0015】また、上記構成によれば、実行アドレス格納手段はマイクロプロセッサの起動時に、最初に行われるプログラムの先頭アドレスがあらかじめ格納されていることにより、マイクロプロセッサの初期化プログラムを簡単に実行することができる。

【0016】また、上記構成によれば、実行アドレス格納手段が通常のデジタルデータの格納手段であるRAM領域内に構成されることによって、メモリーであるRAMを有効に活用することができる。

【0017】また、上記構成によれば、演算手段は実行するプログラムごとに独立したデータ格納領域を備えることによって、実行プログラムごとに、RAM領域内に演算結果の一時的な退避領域を設ける必要がなくなり、RAM領域を有効に使用できる。

【0018】また、上記構成によれば、連続処理開始命令が実行されると、連続処理制御手段により、実行切替禁止信号が出力され、連続処理終了命令が実行されるまでプログラムの実行切替が中止される。

【0019】従って、従来のように、共有しているメモリーや入出力回路に対し、並列して実行している複数のプログラムが全く同時に使用するということが起きず、簡単に誤動作の発生を防止することができる。

【0020】また、上記構成によれば、実行プログラム管理手段が、実行するプログラムのプログラムアドレスデータを実行アドレス格納手段に追加すると、実行状態フラグの該当するフラグが設定され、実行プログラム管理手段が、実行が終了するプログラムのプログラムアド

10

20

30

40

50

レスデータを削除すると、実行状態フラグの該当するフラグの設定が解除される。

【0021】従って、ソフトウェア的に複数のプログラムを実行する従来のタスクモニタ等のように、複数のプログラムの実行状態の管理を行うための複雑なプログラム処理が不用となり、簡単なプログラムで並列に実行しているプログラムの実行状態の管理を行うことができる。

【0022】

【実施例】以下、本発明の一実施例について、図面を参照しながら説明する。

【0023】図1は、本発明の第1の実施例のマイクロプロセッサの構成を示すブロック図である。図1に示すように、マイクロプロセッサは、プログラム格納手段となる命令ROM300と、命令選択手段となる命令選択回路400と、読み出し命令格納領域として命令レジスタPR0~PR9を有する命令レジスタ群500と、命令レジスタを選択する命令レジスタ選択回路510と、命令デコーダ600と、実行アドレス格納領域としてインストラクションポインタIP0~IP9を有する実行アドレス格納手段となる実行アドレス格納回路800と、実行切手段となる実行切回路200と、タイミングジェネレータ100と、実行プログラム選択手段となるプログラム選択回路900と、アドレス更新手段となるアドレス更新回路1000と、分岐アドレス設定手段となる分岐アドレス設定回路1100と、実行プログラム管理手段となる実行プログラム管理回路700と、演算手段となるALU1300、レジスタA1310、レジスタB1320、アキュムレータ1330、フラグ1340と、入出力回路1200と、RAM1400と、RAMの格納領域の何れかを選択するアドレスデコーダ1410、スタックポインタ1420と、データバス1500とアドレスバス1600とからなる。

【0024】タイミングジェネレータ100は外部クロック入力端子10から供給されるクロック信号をもとに命令の実行サイクルの基準となるタイミング信号を発生し、その出力信号は実行切回路200及びマイクロプロセッサ各部に供給されている。実行切回路200は並列に実行しているプログラム数に応じて、プログラムの実行切信号を発生し、その出力信号をプログラム選択回路900及び命令レジスタ選択回路510に供給する。

【0025】命令ROM300は命令群で構成された複数のプログラムが格納されている。命令選択回路400は供給されるプログラムアドレスデータに応じて命令ROM300に格納された命令コードデータの何れかを選択し、選択した命令コードデータが命令の場合には命令レジスタ選択回路510に供給し、選択した命令コードデータがアドレスデータの場合にはアドレスバス1600、分岐アドレス設定回路1100及び実行プログラム

管理回路700に供給し、選択した命令コードデータが数値データの場合にはデータバス1500に供給する。

【0026】命令レジスタ群500は実行アドレス格納回路800のインストラクションポインタIP0~IP9と一対一に対応した命令レジスタPR0~PR9で構成されている。

【0027】命令レジスタ選択回路510は実行切回路200から供給される実行切信号に従って、命令レジスタPR0~PR9の中から何れか1つを選択し、選択した命令レジスタに格納された命令コードデータを命令デコーダ600に供給すると共に、命令選択回路400から供給される新たな命令コードデータをこの命令レジスタに格納する。

【0028】命令デコーダ600は命令レジスタ選択回路510から供給される命令を解読し、マイクロプロセッサ各部に制御信号を供給する。

【0029】実行アドレス格納回路800は実行しているプログラムの実行アドレスを格納する複数のインストラクションポインタIP0~IP9で構成されている。

【0030】プログラム選択回路900は実行切回路200から供給される実行切信号に従って、実行アドレス格納回路800のインストラクションポインタIP0~IP9の中から何れか1つを選択し、選択したインストラクションポインタに格納されたプログラムアドレスデータを命令選択回路400及びアドレス更新回路1000に供給すると共に、アドレス更新回路1000あるいは分岐アドレス設定回路1100から供給されるプログラムアドレスを新たなアドレスとして、このインストラクションポインタに格納する。

【0031】実行プログラム管理回路700は実行された命令が処理の起動命令の場合には、命令デコーダ600からの制御信号によって、命令選択回路400から供給される、新たに起動するプログラムの先頭アドレスデータを実行アドレス格納回路800のインストラクションポインタIP0~IP9の中の未使用のインストラクションポインタに格納すると共に、実行切回路200に実行プログラムの追加指令信号を供給する。また、実行された命令が処理の終了命令の場合には、命令デコーダ600からの制御信号によって、終了するプログラムのプログラムアドレスデータを格納したインストラクションポインタを未使用とすると共に、実行切回路200に実行プログラムの削除指令信号を供給する。

【0032】アドレス更新回路1000はプログラム選択回路900から供給されたプログラムアドレスデータを更新し、更新したプログラムアドレスデータを内部のレジスタに格納すると共に、プログラム選択回路900に供給する。

【0033】分岐アドレス設定回路1100は、実行された命令が分岐命令の場合には、命令選択回路400から供給される分岐アドレスデータをプログラム選択回路

900に供給する。この時、命令デコーダ600からの制御信号によって、プログラム選択回路900はアドレス更新回路1000から供給される更新アドレスデータではなく、分岐アドレス設定回路1100から供給される分岐アドレスデータを実行アドレス格納回路800の該当するインストラクションポインタに格納する。

【0034】ALU1300はデジタルデータの算術および論理演算を実行する演算器であり、データバス1500からレジスタA1310及びレジスタB1320に供給されたデジタルデータの演算を行い、その演算結果データをアキュムレータ1330に供給すると共に演算結果に応じてフラグ1340を変化させる。また、アキュムレータ1330及びフラグ1340に格納されたデータは、演算結果の読み出し命令を実行することによって、データバス1500に供給される。

【0035】RAM1400はデータバス1500を介してデジタルデータの読み書きを行うランダムアクセスメモリであり、アドレスデコーダ1410あるいはスタックポインタ1420の何れかが選択した格納領域に対し、デジタルデータの書き込みあるいは読み出しが行われる。

【0036】アドレスデコーダ1410はアドレスバス1600から供給されるアドレスデータに応じてRAM1400の格納領域の一部を選択する。

【0037】スタックポインタ1420はRAM1400の格納領域の1つを選択するポインタレジスタとそのポインタレジスタに格納されたデータを1つつ増加あるいは減少させる加減算器で構成され、命令デコーダ600から供給される制御信号によって、ポインタレジスタのデータが増加あるいは減少する。

【0038】I/Oポート1200はマイクロプロセッサの外部とのデータの入出力を行う入出力ポートであり、命令デコーダ600から供給される制御信号に従って、データバス1500から供給されるデジタルデータを出力端子00~015より出力あるいは、入力端子10~115から入力されるデジタルデータをデータバス1500に供給する。

【0039】以上のように構成された第1の実施例のマイクロプロセッサについて、図1、図2及び図3を用いてその動作を説明する。

【0040】図2及び図3は図1に示すマイクロプロセッサの主要部の入出力信号のタイミングチャートである。

【0041】図2及び図3において、aは外部クロック入力端子10から供給されるクロック信号を示し、b0~b3はタイミングジェネレータ100が出力する命令の実行サイクルの基準信号を示し、c0~c2は実行切換回路200が出力するプログラムの実行切換信号を示し、d0~d2は実行アドレス格納回路800のインストラクションポインタIP0~IP2の格納データを示

し、eはプログラム選択回路900が出力する実行プログラムアドレスデータを示し、fはアドレス更新回路1000からプログラム選択回路900に供給される更新アドレスデータを示す。

【0042】以下の説明で、プログラム選択回路900が実行アドレス格納回路800のインストラクションポインタIP0に格納されたプログラムアドレスデータを出力し、また、命令レジスタ選択回路510が命令レジスタPR0に対し、格納されている命令コードデータの出力及び、新たな命令コードデータの格納を行うことによって、1つのプログラムが実行される処理サイクルを処理サイクル0と呼び、同様に、プログラム選択回路900が実行アドレス格納回路800のインストラクションポインタIP1~IP9に格納されたプログラムアドレスデータを出力し、また、命令レジスタ選択回路510が命令レジスタPR1~PR9に対し、格納されている命令コードデータの出力及び、新たな命令コードデータの格納を行うことによって、1つのプログラムが実行される処理サイクルをそれぞれ処理サイクル1~処理サイクル9と呼ぶことにする。

【0043】まず図2を用いて、1つのプログラムのみを実行している状態から、並列して複数のプログラムの実行を行う状態に移行する場合の動作について説明する。

【0044】初期状態として、実行しているプログラムは1つだけであり、この実行しているプログラムをプログラム0とする。従って図2の時刻t0の時点では、インストラクションポインタIP0にはプログラム0の実行位置を示すアドレスデータA0、その他のインストラクションポインタIP1~IP9は未使用であるため、それぞれデータとして、'0'が格納されているとする。また、命令レジスタPR0~PR9にも初期値として、それぞれデータ'0'が格納されているとする。

(命令コードデータが'0'の場合は、マイクロプロセッサの動作としてはなにも行われず、実行時間のみが経過するものとする。) タイミングジェネレータ100は外部クロック入力端子10からクロック信号aが供給されることによって、命令の実行サイクルの基準信号b0~b3を発生し、この信号を実行切換回路200及びマイクロプロセッサの各部に供給する。実行切換回路200はこの信号を基準としてプログラムの実行切換信号を発生する。

【0045】実行切換回路200はタイミングジェネレータから供給される信号b0のリーディングエッジのタイミングで、それぞれ処理サイクル0~処理サイクル9の実行サイクルとなる実行切換信号c0~c9(図2ではc3~c9は図示していない)を発生する。

【0046】図2で、時刻t8までは実行しているプログラムは、処理サイクル0によって実行されているプログラム0だけであり、このプログラムの実行を有効とす

10

20

30

40

50

る信号c 0の電位が高電位レベルとなり、信号c 1～c 9の電位は低電位レベルとなる。(以下、信号の電位は必ず、高電位レベルと低電位レベルのどちらかの状態となることとし、高電位レベルの場合を信号のイネーブル状態と記述する。) 図2で、第1実行サイクル(時刻t 0～時刻t 4の期間)では、信号c 0がイネーブル状態であるので、処理サイクル0即ち、プログラム0の実行サイクルとなる。

【0047】信号c 0がイネーブル状態となることによって、プログラム選択回路900は実行アドレス格納回路800のインストラクションポインタIP0を選択する。さらに、信号c 0及び信号b 3の両方の信号がイネーブル状態となっている期間即ち、時刻t 0～t 2の期間では、プログラム選択回路900は図2eに示すように、インストラクションポインタIP0に格納されたプログラムアドレスデータA0を命令選択回路400及びアドレス更新回路1000に供給する。

【0048】命令選択回路400は命令ROM300に格納された命令群の中から、プログラム選択回路900より供給されるプログラムアドレスデータA0に対応する命令を選択し、命令レジスタ選択回路510に供給する。命令レジスタ選択回路510は実行切換回路200から供給される実行切換信号c 0がイネーブル状態となることによって、インストラクションポインタIP0に対応した命令レジスタPR0を選択し、この命令レジスタPR0に初期値として格納されている命令コード‘0’を命令デコーダ600に供給する。命令デコーダ600は命令コードが‘0’であるため、制御信号の出力は行わない。

【0049】命令レジスタ選択回路510は時刻t 1で信号b 1のリーディングエッジが到来すると、命令選択回路400から供給される命令コードデータをこの命令レジスタPR0に格納する。

【0050】アドレス更新回路1000は図2fに示すように、プログラム選択回路900から供給されるアドレスデータA0を更新したアドレスデータA0+1を内部のレジスタに格納し、第1実行サイクルの期間中、プログラム選択回路900に供給し続ける。プログラム選択回路900は、時刻t 2で信号b 3がイネーブル状態でなくなると、インストラクションポインタIP0に格納されたプログラムアドレスデータA0の出力を中止し、時刻t 3で信号b 1のトレイリングエッジが到来すると、アドレス更新回路1000から供給される更新したアドレスデータA0+1をインストラクションポインタIP0に格納する。以上の一連の動作により、プログラム0の実行サイクルである第1実行サイクルが終了する。

【0051】第2実行サイクル(時刻t 4～時刻t 7の期間)では、信号c 0がイネーブル状態であるので、引き続きプログラム0の実行サイクルとなる。

【0052】時刻t 4で信号b 3がイネーブル状態となることによって、プログラム選択回路900はインストラクションポインタIP0に格納されたプログラムアドレスデータA0+1を命令選択回路400及びアドレス更新回路1000に供給し、命令レジスタ選択回路510はプログラムレジスタPR0に格納されている命令を命令デコーダ600に供給する。

【0053】この命令がプログラム1の実行を起動するための処理起動命令の場合には、命令デコーダ600から供給される制御信号に従って、命令選択回路400は命令ROM300に格納された命令群の中からプログラムアドレスデータA0+1に対応する命令コードデータ、即ちこの場合には起動プログラムであるプログラム1の先頭アドレスデータA1を選択し、実行プログラム管理回路700に供給する。実行プログラム管理回路700は、実行切換回路200に実行プログラム追加信号を供給すると共に、時刻t 5で信号b 2のトレイリングエッジが到来すると、実行アドレス格納回路800の中で未使用であるインストラクションポインタIP1に命令選択回路400から供給された起動プログラムの先頭アドレスデータA1を格納する。

【0054】時刻t 6で信号b 1のトレイリングエッジが到来すると、時刻t 3の場合と同様に、プログラム選択回路900はアドレス更新回路1000から供給された更新アドレスデータA0+2をインストラクションポインタIP0に格納する。

【0055】第2実行サイクルの実行命令が処理起動命令であり、第2実行サイクルで命令選択回路400が選択した命令は新たに起動されるプログラム1の先頭アドレスであるため、命令レジスタPR0には命令レジスタ選択回路510によって命令コード‘0’が格納される。

【0056】第3実行サイクル(時刻t 7～時刻t 8の期間)では、信号c 0がイネーブル状態であるので、プログラム0の実行サイクルとなり、命令レジスタ選択回路510から命令レジスタPR0に格納された命令コード‘0’が命令デコーダ600に供給される。命令コード‘0’はマイクロプロセッサの動作としてはなにも行われないため、第1実行サイクルの場合と同様にし、インストラクションポインタIP0に格納されたプログラムアドレスの更新と命令レジスタPR0への新たな命令の格納のみが行われる。

【0057】第4実行サイクル(時刻t 8～時刻t 12の期間)では、第2実行サイクルで実行された処理起動命令により、処理サイクル1の実行サイクルとなり、プログラム1の実行を有効とする信号c 1がイネーブル状態となる。

【0058】信号c 1がイネーブル状態となることによって、プログラム選択回路900は実行アドレス格納回路800のインストラクションポインタIP1を選択す

10

20

30

40

50



る。さらに、信号c1及び信号b3の両方の信号がイネーブル状態となっている期間即ち、時刻t8～t10の期間では、プログラム選択回路900は図2eに示すように、インストラクションポインタIP1に格納されたプログラムアドレスデータA1を命令選択回路400及びアドレス更新回路1000に供給する。

【0059】命令選択回路400は命令ROM300に格納された命令群の中から、プログラム選択回路900より供給されるプログラムアドレスデータA1に対応する命令を選択し、命令レジスタ選択回路510に供給する。命令レジスタ選択回路510は実行切換回路200から供給される実行切換信号c1がイネーブル状態となることによって、インストラクションポインタIP1に対応した命令レジスタPR1を選択し、命令レジスタPR1に格納されている命令コード‘0’を命令デコーダ600に供給する。命令デコーダ600は命令コードが‘0’であるため、制御信号の出力は行わない。

【0060】命令レジスタ選択回路510は時刻t9で信号b1のリーディングエッジが到来すると、命令選択回路400から供給される命令を命令レジスタPR1に格納する。

【0061】アドレス更新回路1000はプログラム選択回路900から供給されるアドレスデータA1を更新したアドレスデータA1+1をプログラム選択回路900に供給する。プログラム選択回路900は、時刻t10で信号b3がイネーブル状態でなくなると、インストラクションポインタIP1に格納されたプログラムアドレスデータA1の出力を中止し、時刻t11で信号b1のトレイリングエッジが到来すると、アドレス更新回路1000から供給される更新したアドレスデータA1+1をインストラクションポインタIP1に格納する。以上の動作により、処理サイクル1、即ち、プログラム1の実行サイクルである第4実行サイクルが終了する。

【0062】第5実行サイクル（時刻t12～時刻t15の期間）では、信号c0がイネーブル状態となるため、プログラム0の実行サイクルとなる。

【0063】第5実行サイクルでは第2実行サイクルの場合と同様に、プログラム2の実行を起動するための処理起動命令が行われる。即ち、命令デコーダ600から供給される制御信号に従って、実行プログラム管理回路700は、命令選択回路400から供給される起動プログラムの先頭アドレスデータA2を時刻t13で、実行アドレス格納回路800の未使用であるインストラクションポインタIP2に格納すると共に、実行切換回路200に実行プログラム追加信号を供給する。

【0064】さらに、時刻t14では時刻t6の場合と同様に、アドレス更新回路1000から供給された更新アドレスデータA0+4がプログラム選択回路900によって、インストラクションポインタIP0に格納される。

【0065】また、第5実行サイクルの実行命令が処理起動命令であり、第5実行サイクルで命令選択回路400が選択した命令は新たに起動されるプログラム2の先頭アドレスであるため、命令レジスタPR0には命令レジスタ選択回路510によって命令コードとして‘0’が格納される。

【0066】第6実行サイクル（時刻t15～時刻t19の期間）では、信号c1がイネーブル状態であるので、プログラム1の実行サイクルとなり、命令レジスタ選択回路510から命令レジスタPR1に格納された命令コードデータが命令デコーダ600に供給される。

【0067】この命令が分岐命令の場合には、命令デコーダ600から供給される制御信号に従って、命令選択回路400は命令ROM300に格納された命令群の中からプログラムアドレスデータA1+1に対応する命令、即ちこの場合には分岐アドレスデータB1を選択し、分岐アドレス設定回路1100に供給する。

【0068】時刻t16で信号b2のトレイリングエッジが到来すると、プログラム選択回路900は分岐アドレス設定回路1100から供給される分岐アドレスデータB1を実行アドレス格納回路800のインストラクションポインタIP1に格納する。

【0069】第6実行サイクルの実行命令が分岐命令であり、第6実行サイクルで命令選択回路400が選択した命令は分岐アドレスであるため、時刻t17で信号b1のリーディングエッジが到来すると、命令レジスタPR1には命令レジスタ選択回路510によって命令コード‘0’が格納される。

【0070】また、分岐命令が実行されたため、命令デコーダ600から供給される制御信号によって、時刻t18で信号b1のトレイリングエッジが到来しても、プログラム選択回路900はアドレス更新回路1000から供給された更新アドレスデータをインストラクションポインタIP2に格納する動作は行わない。

【0071】第7実行サイクル（時刻t19～時刻t22の期間）では、第5実行サイクルで実行された処理起動命令により、処理サイクル2の実行サイクルとなり、プログラム2の実行を有効とする信号c2がイネーブル状態となる。

【0072】信号c2がイネーブル状態となることによって、プログラム選択回路900は実行アドレス格納回路800のインストラクションポインタIP2を選択する。さらに、信号c2及び信号b0の両方の信号がイネーブル状態となっている期間では、プログラム選択回路900は図2eに示すように、インストラクションポインタIP1に格納されたプログラムアドレスデータA2を命令選択回路400及びアドレス更新回路1000に供給する。

【0073】命令選択回路400は命令ROM300に格納された命令群の中から、プログラム選択回路900

10

20

30

40

50

より供給されるプログラムアドレスデータA2に対応する命令を選択し、命令レジスタ選択回路510に供給する。命令レジスタ選択回路510は実行切替回路200から供給される実行切替信号c2がイネーブル状態となることによって、インストラクションポインタIP2に対応した命令レジスタPR2を選択し、命令レジスタPR2に格納されている命令コード'0'を命令デコーダ600に供給する。命令デコーダ600は命令コードが'0'であるため、制御信号の出力は行わない。

【0074】命令レジスタ選択回路510は時刻t20で信号b1のリーディングエッジが到来すると、命令選択回路400から供給される命令を命令レジスタPR2に格納する。

【0075】アドレス更新回路1000はプログラム選択回路900から供給されるアドレスデータA2を更新したアドレスデータA2+1をプログラム選択回路900に供給する。プログラム選択回路900は、信号b3がイネーブル状態でなくなると、インストラクションポインタIP1に格納されたプログラムアドレスデータA2の出力を中止し、時刻t21で信号b1のトレイリングエッジが到来すると、アドレス更新回路1000から供給される更新したアドレスデータA2+1をインストラクションポインタIP2に格納する。以上の動作により、処理サイクル2即ち、プログラム2の実行サイクルである第7実行サイクルが終了する。

【0076】以下第8～第10実行サイクルでは信号c0, c1, c2が交互にイネーブル状態となり、順番に処理サイクル0～処理サイクル2となるため、プログラム0～プログラム2が順次実行される。

【0077】以上のようにして、図2の第2, 第5実行サイクルで処理起動命令が実行され、実行プログラム管理回路700が、それまで実行アドレス格納回路800の未使用領域であったインストラクションポインタIP1及びIP2にそれぞれプログラム1, プログラム2の先頭アドレスデータであるA1及びA2を格納し、実行切替回路200に実行プログラムの追加指令信号を供給することによって、第4, 第7実行サイクルから処理サイクル1, 処理サイクル2が有効となり、それぞれ、プログラム1, プログラム2の実行が開始される。そして、第8実行サイクル以降では、処理サイクル0～処理サイクル2が順番に行われるため、3つのプログラム、即ち、プログラム0～プログラム2が時分割で並列に実行される。

【0078】実行アドレス格納回路800は10個のインストラクションポインタ(IP0～IP9)で構成されているので、同様に、並列に10個のプログラムまで実行することが可能である。

【0079】次に図3を用いて、3つのプログラムを並列に実行している状態から、1つのプログラムのみが実行している状態に移行する場合の動作について説明す

る。

【0080】図3で時刻t0の時点では、処理サイクル0～処理サイクル2が順番に到来し、それぞれの処理サイクルでプログラム0～プログラム2の3つのプログラムが実行されている。この実行されているプログラムアドレスデータはA0'～A2'であり、それぞれインストラクションポインタIP0～IP2に格納されており、その他のインストラクションポインタIP3～IP9は未使用であるため、それぞれデータとして、'0'が格納されているとする。

【0081】時刻t0～t8の期間では、図2で説明したようにプログラム0～プログラム2が順次実行されている。

【0082】図3の第5実行サイクル(時刻t4～時刻t6の期間)では、信号c1がイネーブル状態であるので、処理サイクル1、即ち、プログラム1の実行サイクルである。この第5実行サイクルで実行される命令が処理終了命令の場合、命令デコーダ600から供給される制御信号によって、実行プログラム管理回路700は実行切替回路200に実行プログラム削除信号を供給すると共に、時刻t5で信号b2のトレイリングエッジが到来すると、プログラム1のプログラムアドレスデータを格納しているインストラクションポインタIP1にデータ'0'を格納し、未使用状態とする。

【0083】実行切替回路200は実行プログラム削除信号が供給されると、第5実行サイクル終了後は、信号c1をイネーブル状態にせず、従って、処理サイクル1の実行時間の割り当てを行わないようにする。

【0084】また、第5実行サイクルの実行命令が処理終了命令であるため、命令デコーダ600から供給される制御信号によって、命令レジスタ選択回路510は命令レジスタPR1に命令コード'0'を格納し、プログラム選択回路900はアドレス更新回路1000から供給された更新アドレスデータをインストラクションポインタIP1に格納する動作を中止する。

【0085】同様にして、処理サイクル2の実行サイクルである第6実行サイクル(時刻t6～時刻t8の期間)で、処理終了命令が実行されると、実行プログラム管理回路700は実行切替回路200に実行プログラム削除信号を供給すると共に、時刻t7で信号b2のトレイリングエッジが到来すると、プログラム2のプログラムアドレスデータを格納しているインストラクションポインタIP2にデータ'0'を格納し、未使用状態とする。

【0086】実行切替回路200は実行プログラム削除信号が供給されることによって、第6実行サイクル終了後は、信号c2をイネーブル状態にせず、処理サイクル2の実行時間の割り当てを行わないようにする。

【0087】また、命令レジスタ選択回路510は命令レジスタPR2に命令コード'0'を格納し、プログラ

ム選択回路900はアドレス更新回路1000から供給された更新アドレスデータをインストラクションポインタIP2に格納する動作を中止する。

【0088】第5、第6実行サイクルで処理終了命令が実行され、処理サイクル1及び処理サイクル2の実行時間の割り当てが行われなくなる。従って、第7実行サイクル以降は信号c0のみがイネーブル状態となり、処理サイクル0が連続し、プログラム0だけが実行されるようになる。

【0089】以上図2、図3を用いて説明したようにして、並列に実行するプログラムの追加及び削除を行うことができる。

【0090】このように第1の実施例では、マイクロプロセッサで並列に実行するプログラムを増加させる場合には、処理起動命令を実行することにより、プログラム管理回路700は、新たに実行されるプログラムの先頭アドレスデータを実行アドレス格納回路800のインストラクションポインタIP0～IP9の中の未使用であるインストラクションポインタに格納すると共に、実行切替回路200に実行プログラム追加信号を供給する。そして、実行切替回路200は並列に実行するプログラム数に応じて実行切替信号を発生する。また、並列に実行するプログラムを減少させる場合には、処理終了命令を実行することによって、プログラム管理回路700は、実行を終了させるプログラムのアドレスデータが格納されたインストラクションポインタのデータ‘0’を格納し、未使用状態とし、実行切替回路200に実行プログラム削除信号を供給する。これによって、実行切替回路200は、実行が終了したプログラムの実行サイクルとなっていた処理サイクルの実行時間の割り当てを行わないようにする。

【0091】即ち、処理起動命令の実行で、新たに実行するプログラムのアドレスデータがインストラクションポインタIP0～IP9に格納され、増加したプログラム数に応じた実行切替信号が発生され、処理終了命令の実行で、インストラクションポインタIP0～IP9に格納されたアドレスデータが削除され、減少したプログラム数に応じた実行切替信号が発生されることにより、行うべき処理数にあわせて、並列に実行されるプログラム数を容易に変更することができる。

【0092】その結果、独立に、しかも非同期に発生する複数の処理に対して即座に対処することができるなお第1の実施例では、実行アドレス格納回路800は10個のインストラクションポインタ(IP0～IP9)で構成されているので、並列に実行できるプログラムは最大10個までであるが、実行アドレス格納回路800のインストラクションポインタの個数と実行切替回路から出力する実行切替信号の本数を増減させることによって、並列に実行するプログラム数はいくつの場合でも同様に実現できる。また、第1の実施例では、処理起

動命令や分岐命令は2回の実行サイクルで処理が行われるようになっているが、命令レジスタ群500(PR0～PR9)や命令レジスタ選択回路510をなくし、命令コードの長さを変えることにより、1回の実行サイクルで処理を行うようにした場合でも、処理の追加、削除の動作は全く同様にして実現できる。

【0093】さらに、第1の実施例では処理起動命令が実行された実行サイクルから2つ目の実行サイクルで、新たなプログラムが起動されるように説明したが、実行切替回路200から供給する実行切替信号の出力タイミングを変更し、処理起動命令が実行された実行サイクルの次の実行サイクルから、新たなプログラムが起動するようにしても全く同様に、並列して実行するプログラムの追加を行うことができる。

【0094】第1の実施例では、処理サイクル0で実行しているプログラム0から処理起動命令を実行し、処理サイクル1、処理サイクル2の処理が有効となる場合について説明しているが、処理サイクル0からだけではなく、すでに有効となった処理サイクルであれば、その処理サイクルで実行されるプログラムで処理起動命令を実行することによって、有効となっていない処理サイクルを有効にし、新たなプログラムをその処理サイクルで実行させることができる。

【0095】第1の実施例で、実行アドレス格納回路800のインストラクションポインタの何れか1つ、例えば、IP0にはマイクロプロセッサの起動時に、あらかじめ決められたプログラムアドレスAinが自動的に設定されるように実行アドレス格納回路800を構成し、さらに実行切替回路は起動時には、必ず信号c0をイネーブルにするように構成することによって、命令ROM300のアドレスAinに対応する格納領域に初期プログラムを格納しておけば、マイクロプロセッサの初期状態では常に1つのプログラムが起動できるようになる。この初期プログラムの中で、マイクロプロセッサの初期設定や以後起動するプログラムのためのデータ等の初期設定を行うことによって、簡単に初期化を行うことができる。

【0096】第1の実施例では、実行アドレス格納回路800は独立したアドレスデータ格納回路として構成しているが、インストラクションポインタIP0～IP9をRAM1400のデータ格納領域内の一部の領域で実現し、実行プログラム管理回路700及びプログラム選択回路900が、RAM1400内部のインストラクションポインタとして割り当てられた格納領域に対して動作を行うように構成すしても全く同様の動作が実現できる。

【0097】第1の実施例では、データの演算のために使用するレジスタA1310、レジスタB1320、アキュムレータ1330、フラグ1340はそれぞれ1つであるが、これら全てを処理サイクルごとに独立に設け

10

20

30

40

50

るように構成することによって、並列にデジタルデータの算術および論理演算を容易に実行することが可能なマイクロプロセッサを実現することができる。

【0098】図4は、本発明の第2の実施例のマイクロプロセッサの構成を示すブロック図である。図4では、図1で示される第1の実施例であるマイクロプロセッサと同一の構成要素については図1と同じ符号で示している。第1の実施例との構成の差異は、連続処理制御手段となる連続処理制御回路2000を新たに設けた点である。連続処理制御回路2000は、実行された命令が連続処理開始命令の場合には、この命令を実行したプログラムが連続して処理の実行を行うように、実行切替禁止信号を連続処理終了命令が実行されるまで、実行切替回路200に供給する。実行切替回路200は、連続処理制御回路2000から実行切替禁止信号が供給されている期間には、プログラムの実行切替を行わない。

【0099】以上のように構成された第2の実施例のマイクロプロセッサについて、図4及び図5を用いてその動作を説明する。

【0100】図5は図4に示すマイクロプロセッサの主要部の入出力信号のタイミングチャートである。

【0101】図5において、aは外部クロック入力端子10から供給されるクロック信号を示し、b0~b3はタイミングジェネレータ100が出力する命令の実行サイクルの基準信号を示し、c0~c2は実行切替回路200が出力するプログラムの実行切替信号を示し、d0~d2は実行アドレス格納回路800のインストラクションポインタIP0~IP2の格納データを示し、eはプログラム選択回路900が出力する実行プログラムアドレスデータを示し、fはアドレス更新回路1000からプログラム選択回路900に供給される更新アドレスデータを示し、gは連続処理制御回路2000が実行切替回路200に供給する実行切替禁止信号を示す。

【0102】以下の説明では、第1の実施例で説明したように、プログラム選択回路900が実行アドレス格納回路800のインストラクションポインタIP0~IP9に格納されたプログラムアドレスデータを出力し、また、命令レジスタ選択回路510が命令レジスタPR0~PR9に対し、格納されている命令コードデータの出力及び、新たな命令コードデータの格納を行うことによって、1つのプログラムが実行される処理サイクルをそれぞれ処理サイクル0~処理サイクル9と呼ぶことにする。

【0103】図5で時刻t0の時点では、処理サイクル0~処理サイクル2が順番に到来し、それぞれの処理サイクルでプログラム0~プログラム2の3つのプログラムが実行されている。この実行されているプログラムのプログラムアドレスデータはN0~N2であり、それぞれインストラクションポインタIP0~IP2に格納されており、その他のインストラクションポインタIP3

~IP9は未使用であるため、それぞれデータとして、'0'が格納されているとする。

【0104】時刻t0~t6の間では、第1の実施例で説明したように実行切替回路200から供給される信号c0~c2が順次イネーブル状態となることによって、処理サイクル0~処理サイクル2が順番に到来し、プログラム0~プログラム2が順次実行されている。

【0105】図5の第5実行サイクル(時刻t4~時刻t6の間)では、信号c1がイネーブル状態であるので、処理サイクル1即ち、プログラム1の実行サイクルである。この第5実行サイクルで実行される命令が連続処理開始命令の場合、命令デコーダ600から供給される制御信号によって、図5gに示されるように、時刻t5で連続処理制御回路2000から実行切替回路200に出力される実行切替禁止信号がイネーブル状態となる。実行切替禁止信号がイネーブル状態となることによって、実行切替回路200は実行の切替を中断するため、第6~第8実行サイクルでは、信号c1がイネーブル状態になったままとなり、プログラム1の実行が連続して行われる。

【0106】第8実行サイクル(時刻t8~時刻t10の間)で連続処理終了命令が実行されると、命令デコーダ600から供給される制御信号によって、図5gに示されるように、時刻t9で実行切替禁止信号がイネーブル状態でなくなる。実行切替禁止信号がイネーブル状態でなくなることによって、実行切替回路200は実行の切替を再開し、即ち、第9実行サイクルからは、再び、信号c2、c0、c1の順に実行の切替動作を再開するため、処理サイクル0~処理サイクル2が順番に到来し、プログラム0~プログラム2が順次実行される。

【0107】このように第2の実施例では、連続処理開始命令を実行することによって、連続処理制御回路2000が実行切替禁止信号を出力し、実行切替回路200が実行の切替を中断するため、この連続処理開始命令を実行したプログラムのみが実行を連続的に行うようになる。そして、このプログラムが、連続して実行する一連の命令群の実行を完了し、連続処理終了命令を実行すると、連続処理制御回路2000は実行切替禁止信号の出力を中止するため、このプログラムが連続処理開始命令を実行する前の状態から、実行切替回路200は実行切替信号の出力を再開する。

【0108】即ち、連続処理開始命令を実行した後に、一連の命令群を実行し、その後に、連続処理終了命令を実行することによって、連続して行う処理が複数の命令で構成され、この処理の実行中は他のプログラムの実行を禁止する必要がある一連の処理を容易に行うことができる。

【0109】従って、従来のように複数のプログラムで共有しているメモリーや入出力回路に対し、2つ以上のプログラムが全く同時に使用するということが起きず、

メモリの同一領域や入出力ポートのデータを同時に変更することによって生じるプログラムの誤動作の発生を簡単に防止することができる。

【0110】その結果、マイクロプロセッサの中で共有しているメモリや入出力回路のハードウェアも簡単な構成にすることができる。

【0111】なお、第2の実施例では、処理サイクル1の時に、プログラム1が連続処理開始命令を実行し、プログラム1の実行が連続的に行われる場合について説明しているが、何れの処理サイクルで実行されているプログラムであっても、連続処理開始命令を実行することによって、連続的に実行を続けることが可能である。

【0112】また、第2の実施例では、連続処理開始命令と連続処理終了命令の実行サイクルを除いて、1つのプログラムが2サイクル連続して実行を行う場合について説明しているが、連続処理開始命令を実行してから、連続処理終了命令を実行するまでに、いくつかの命令を実行する場合でも全く同様にして、命令を連続的に実行することが可能である。

【0113】図6は、本発明の第3の実施例のマイクロプロセッサの構成を示すブロック図である。図6では、図1で示される第1の実施例であるマイクロプロセッサと同一の構成要素については図1と同じ符号で示している。第1の実施例との構成の差異は、実行状態フラグ3000を新たに設けた点である。実行状態フラグ3000は、実行アドレス格納回路800の各インストラクションポインタIP0～IP9の使用状況（プログラムアドレスデータが格納されているかどうか）を示す10ビットのフラグレジスタであり、このフラグレジスタは実行プログラム管理回路700によって設定される。即ち、このフラグレジスタはその最下位ビットから順に、各インストラクションポインタIP0～IP9に対するプログラムアドレスの設定の有無を示すフラグとなっており、実行プログラム管理回路700から供給される信号によって、インストラクションポインタにプログラムアドレスデータが設定された場合には、データ‘1’、インストラクションポインタが未使用となった場合には、データ‘0’がこのフラグレジスタの対応するビットに格納される。（以降、実行状態フラグ3000の特定のビットにデータ‘1’あるいはデータ‘0’が格納される場合をそれぞれ、ビットがセットあるいはリセットされると記述する。）また、命令デコーダ600から供給される制御信号により、実行状態フラグ3000のフラグレジスタのデータはデータバス1500に供給される。

【0114】以上のように構成された第3の実施例のマイクロプロセッサについて、図6及び図7を用いてその動作を説明する。

【0115】図7は図6に示すマイクロプロセッサの主要部の入出力信号のタイミングチャートである。

【0116】図7において、aは外部クロック入力端子10から供給されるクロック信号を示し、b0～b3はタイミングジェネレータ100が出力する命令の実行サイクルの基準信号を示し、c0～c2は実行切換回路200が出力するプログラムの実行切換信号を示し、d0～d2は実行アドレス格納回路800のインストラクションポインタIP0～IP2の格納データを示し、eはプログラム選択回路900が出力する実行プログラムアドレスデータを示し、fはアドレス更新回路1000からプログラム選択回路900に供給される更新アドレスデータを示し、hは実行状態フラグ3000の内部のフラグの状態を示す。

【0117】以下の説明では、第1、第2の実施例で説明したように、プログラム選択回路900が実行アドレス格納回路800のインストラクションポインタIP0～IP9に格納されたプログラムアドレスデータを出力し、また、命令レジスタ選択回路510が命令レジスタPR0～PR9に対し、格納されている命令コードデータの出力及び、新たな命令コードデータの格納を行うことによって、1つのプログラムが実行される処理サイクルをそれぞれ処理サイクル0～処理サイクル9と呼ぶことにする。

【0118】図7で時刻t0の時点では、マイクロプロセッサで実行しているプログラムは、処理サイクル0によって実行されているプログラム0だけであり、従ってインストラクションポインタIP0にはアドレスデータA0、その他のインストラクションポインタIP2～IP9は未使用であるため、それぞれデータとして、

‘0’が格納されているとする。また、実行状態フラグ3000のフラグレジスタはプログラム0のプログラムアドレスが格納されるIP0に対応するビット位置、即ち最下位ビットのみがセットされているので、このフラグレジスタのデータは‘1’となっている。

【0119】図7の第2実行サイクル（時刻t1～時刻t3の期間）では、第1の実施例で説明したように実行切換回路200から供給される信号c0がイネーブル状態であるので、処理サイクル0即ち、プログラム0の実行サイクルである。この第2実行サイクルで処理起動命令が実行されると、命令デコーダ600から制御信号が供給され、命令選択回路400は命令ROM300に格納された命令群の中からプログラムアドレスデータA0+1に対応する命令、即ちこの場合には起動プログラムであるプログラム1の先頭アドレスデータA1を選択し、実行プログラム管理回路700に供給する。

【0120】実行プログラム管理回路700は、時刻t2で信号b2のトレイリングエッジが到来すると、実行アドレス格納回路800の中で未使用であるインストラクションポインタIP1に命令選択回路400から供給された起動プログラムの先頭アドレスデータA1を格納すると共に、実行切換回路200に実行プログラム追加

信号を供給し、実行状態フラグ3000のフラグレジスタのインストラクションポインタIP1に対応するビット位置即ち最下位から2ビット目のフラグをセットする。このビットがセットされることによって、フラグレジスタのデータは‘3’となる。

【0121】この第2実行サイクルで処理起動命令が実行されることにより、第4実行サイクルから処理サイクル1が有効となり、新たに起動されたプログラムであるプログラム1の実行が開始される。

【0122】第5実行サイクル（時刻t6～時刻t8の期間）では、信号c0がイネーブル状態であるので、プログラム0の実行サイクルとなる。

【0123】この第5実行サイクルで処理起動命令が実行されると、第2実行サイクルの場合と同様にして、時刻t7で信号b2のトレイリングエッジの到来によって、実行プログラム管理回路700は、実行アドレス格納回路800の中で未使用であるインストラクションポインタIP2に命令選択回路400から供給された起動プログラムであるプログラム2の先頭アドレスデータA2を格納すると共に、実行切換回路200に実行プログラム追加信号を供給し、実行状態フラグ3000のフラグレジスタのインストラクションポインタIP2に対応するビット位置即ち最下位から3ビット目のフラグをセットする。このビットがセットされることによって、フラグレジスタのデータは‘7’となる。

【0124】この第5実行サイクルで処理起動命令が実行されることにより、第7実行サイクルから処理サイクル2が有効となり、新たに起動されたプログラムであるプログラム2の実行が開始される。

【0125】第9実行サイクル（時刻t11～時刻t13の期間）では、信号c1がイネーブル状態となるので、処理サイクル1即ち、プログラム1の実行サイクルとなる。この第9実行サイクルで処理終了命令が実行されると、命令デコーダ600から供給される制御信号に従って、時刻t12で信号b2のトレイリングエッジが到来することで、実行プログラム管理回路700はプログラム1のプログラムアドレスデータを格納しているインストラクションポインタIP1にデータ‘0’を格納し、未使用状態とすると共に、実行切換回路200に実行プログラム削除信号を供給し、実行状態フラグ3000のフラグレジスタのインストラクションポインタIP1に対応するビット位置、即ち最下位から2ビット目のフラグをリセットする。このビットがリセットされることによって、フラグレジスタのデータは‘5’となる。

【0126】以上のように、実行状態フラグ3000のフラグレジスタは処理起動命令及び処理終了命令によって、各ビットがセットあるいはリセットされるため、このフラグレジスタは常にインストラクションポインタIP0～IP9の動作状態を示している。従って、実行状態フラグ3000のフラグレジスタのデータを読み出す

命令を実行し、このフラグレジスタのデータをデータバス1500に供給することによって、実行中のプログラムがさらに処理を起動できるかどうかを判断することが可能となる。

【0127】このように第3の実施例では、処理起動命令が実行されると、実行プログラム管理回路700は、実行を開始するプログラムの先頭アドレスデータを実行アドレス格納回路800の未使用であるインストラクションポインタに格納すると共に、実行状態フラグ3000のフラグレジスタに対し、新たにプログラムの先頭アドレスを格納したインストラクションポインタに対応するビット位置のフラグをセットする。また、処理終了命令が実行されると、実行プログラム管理回路700は、実行を終了するプログラムのプログラムアドレスデータが格納されたインストラクションポインタにデータ

‘0’を格納し、未使用状態とすると共に、実行状態フラグ3000のフラグレジスタに対し、この未使用となったインストラクションポインタに対応するビット位置のフラグをリセットする。

【0128】即ち、実行状態フラグ3000のフラグレジスタは、新たなプログラムが起動されると、そのプログラムを実行するために使用されているインストラクションポインタに対応したビット位置がセットされ、プログラムが終了すると、そのプログラムを実行するために使用されていたインストラクションポインタに対応したビット位置がリセットされるので、この実行状態フラグ3000のフラグレジスタを確認することによって、インストラクションポインタの使用状況つまり実行プログラム数を容易に把握することができる。

【0129】従って、ソフトウェア的に複数のプログラムを実行する従来のタスクモニタ等のように、複数のプログラムの実行状態の管理を行うための複雑なプログラム処理が不要となり、簡単なプログラムで並列に実行しているプログラムの実行状態の管理を行うことができる。

【0130】その結果、このプロセッサではプログラムの実行スケジュール管理等を簡単に行うことができる。

【0131】なお、第3の実施例では実行状態フラグ3000によって、インストラクションポインタの使用状況のみを確認するように説明しているが、実行中のプログラムに対し、特定のプログラムはどのプログラムが実行した処理起動命令によって、起動されたかを確認したり、インストラクションポインタにプログラムアドレスデータは格納されているが、連続処理開始命令等により、一時的に実行が中断されているプログラムを確認することも、実行状態フラグ3000のビット数やフラグレジスタの種類を増やすことにより、全く同様にして実現できる。

【0132】

【発明の効果】以上のように本発明によれば、並列に実

行するプログラム数を増加させる場合には、プログラム管理手段が、実行するプログラムのプログラムアドレスデータを実行アドレス格納手段に追加し、実行切替手段が増加した実行プログラム数に応じて、プログラムの実行切替信号を発生する。また、並列に実行するプログラム数を減少させる場合には、プログラム管理手段が、実行を終了させるプログラムのプログラムアドレスデータを実行アドレス格納手段から削除し、実行切替手段が減少した実行プログラム数に応じて、プログラムの実行切替信号を発生する。

【0133】従って、従来のように、複数のマイクロプロセッサを使用したり、1つのマイクロプロセッサ上で、複数のプログラムの実行の切替を可能にするソフトウェアであるタスクモニタ等を使用することなく、必要な数だけ複数のプログラムを容易に実行することができる。その結果、独立に、しかも非同期に発生する複数の処理に対して即座に対処することができる。

【図面の簡単な説明】

【図1】本発明の第1の実施例のマイクロプロセッサの構成を示すブロック図

【図2】図1に示すマイクロプロセッサで並列に実行するプログラム数を増加させる場合の主要部の入出力信号のタイミングチャート

【図3】図1に示すマイクロプロセッサで並列に実行するプログラム数を減少させる場合の主要部の入出力信号のタイミングチャート

【図4】本発明の第2の実施例のマイクロプロセッサの構成を示すブロック図

【図5】図4に示すマイクロプロセッサの主要部の入出力信号のタイミングチャート

【図6】本発明の第3の実施例のマイクロプロセッサの構成を示すブロック図

【図7】図6に示すマイクロプロセッサの主要部の入出力信号のタイミングチャート

【符号の説明】

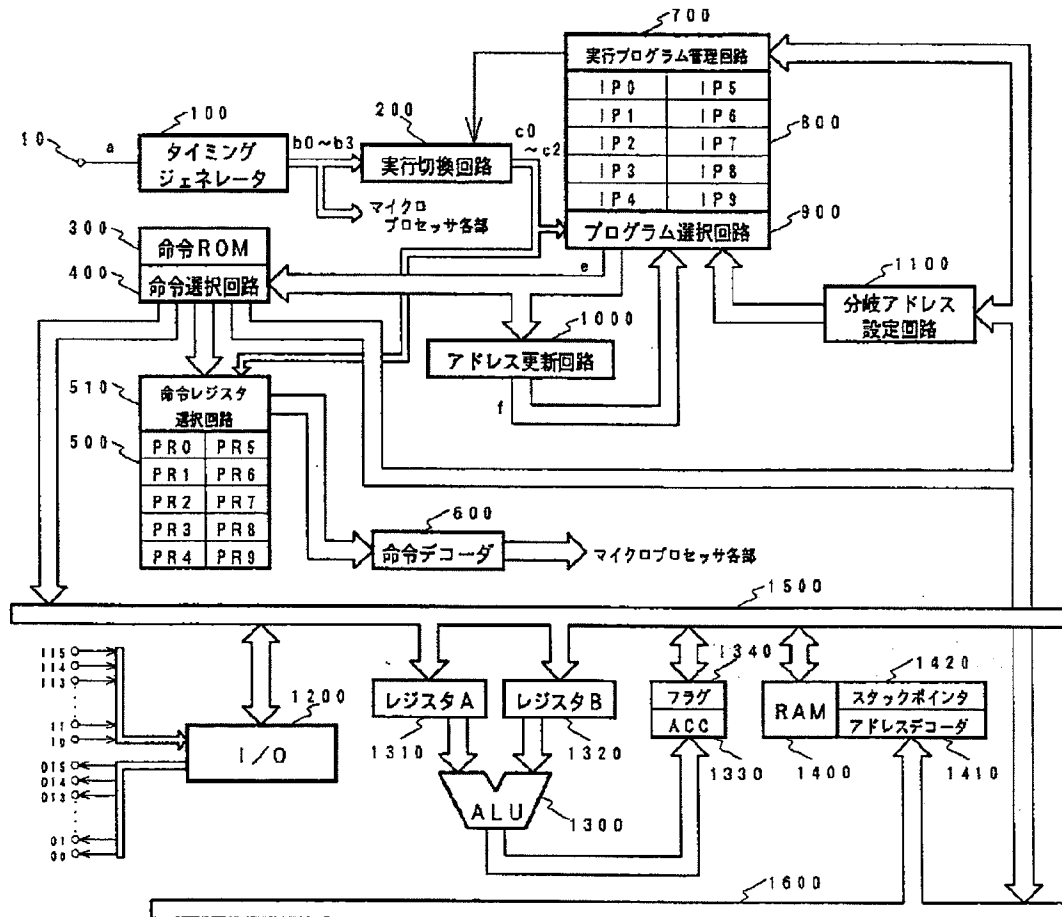
- 10 外部クロック入力端子
- 100 タイミングジェネレータ
- 200 実行切替回路
- 300 命令ROM
- 400 命令選択回路
- 500 命令レジスタ群
- 510 命令レジスタ選択回路
- 600 命令デコーダ
- 700 実行プログラム管理回路
- 800 実行アドレス格納回路
- 900 プログラム選択回路
- 1000 アドレス更新回路
- 1100 分岐アドレス設定回路
- 1200 I/Oポート
- 1300 ALU
- 1310 レジスタA
- 1320 レジスタB
- 1330 アキュムレータ
- 1340 フラグ
- 1400 RAM
- 1410 アドレスデコーダ
- 1420 スタックポインタ
- 1500 データバス
- 1600 アドレスバス
- 2000 連続処理制御回路
- 3000 実行状態フラグ
- PR0~PR9 命令レジスタ
- 00~015 出力端子
- I0~I15 入力端子

10

20

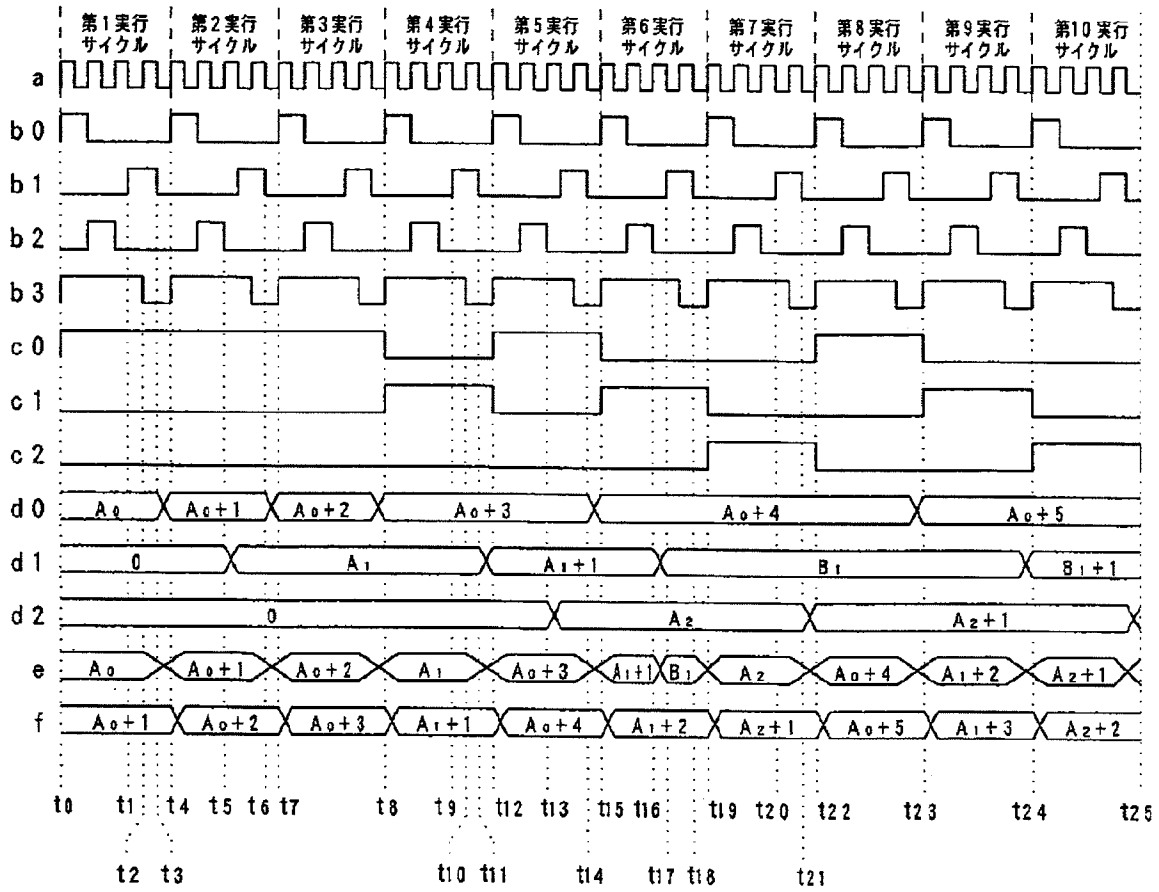
30

【図1】

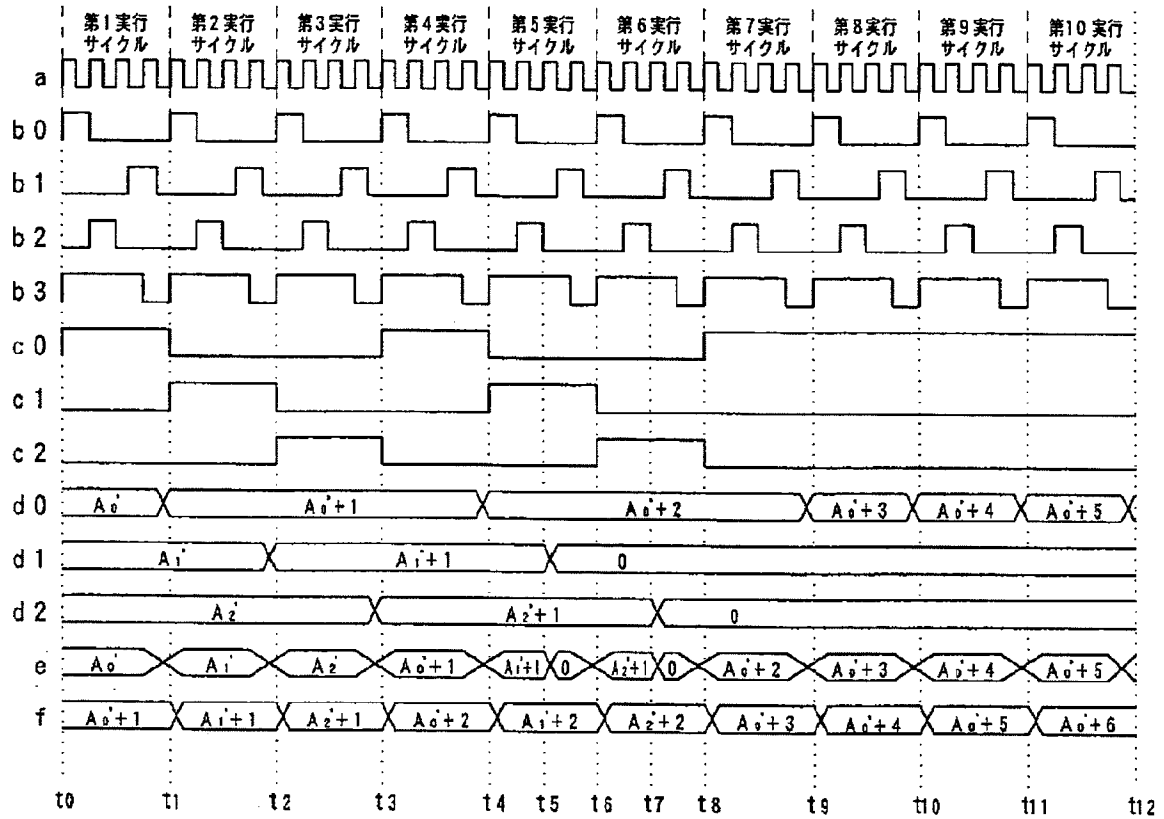




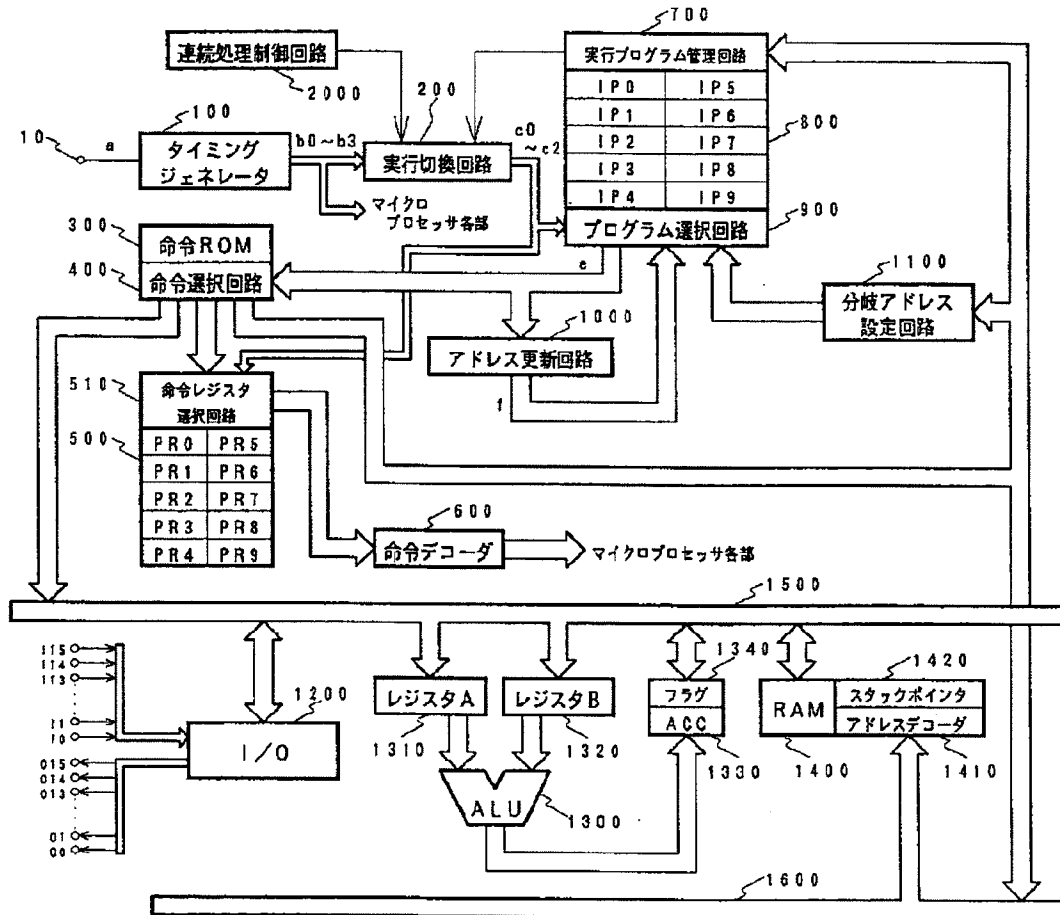
【図2】



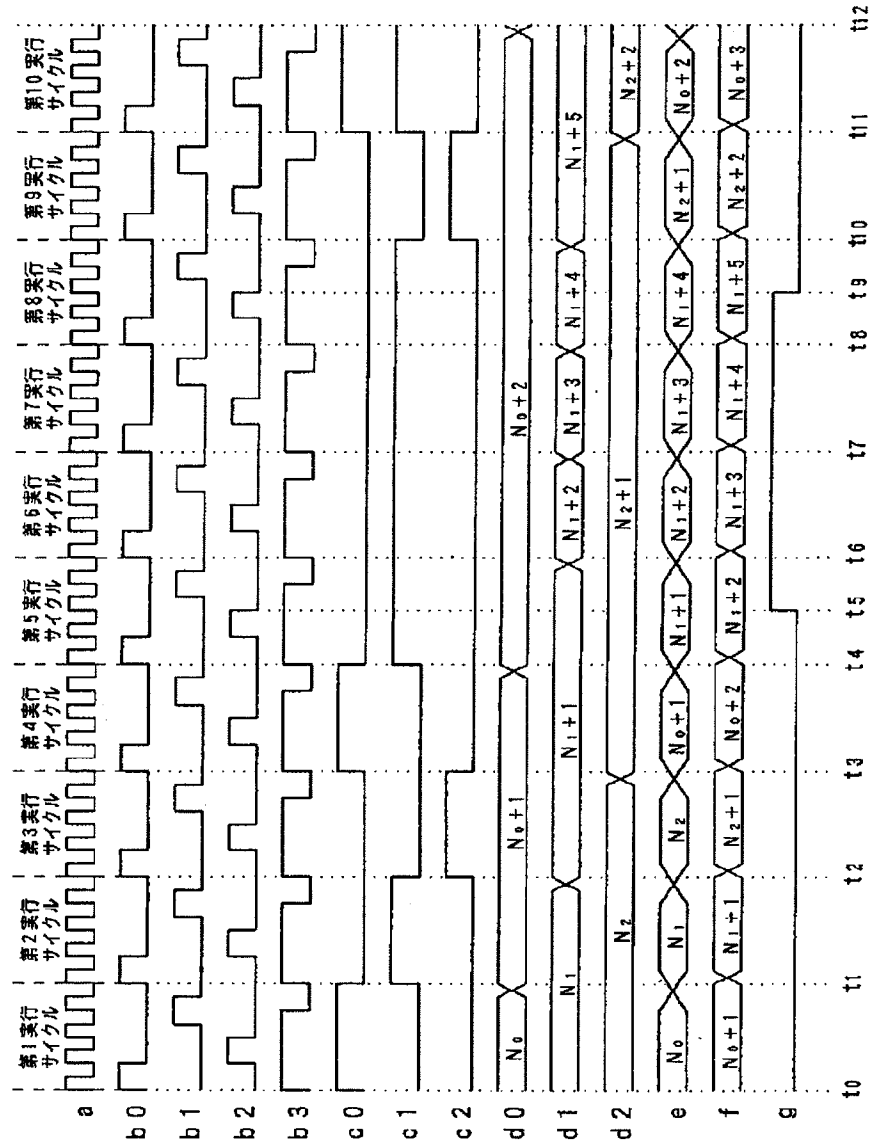
【図3】



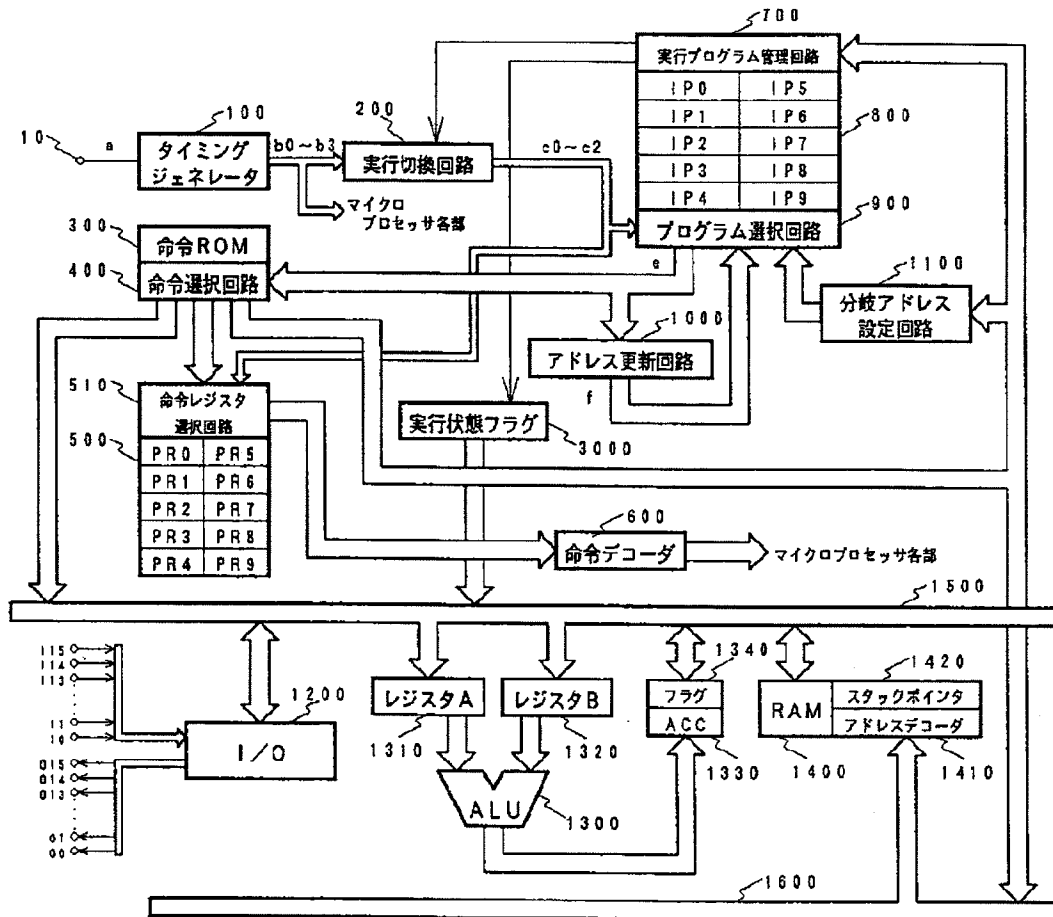
【図4】



【図5】



【図6】



【図7】

